BOOK REVIEWS

J. New York Entomol. Soc. 97(2):234–241, 1989

CLADISTICS IN THE FAST LANE

Hennig86. Version 1.5.—J. S. Farris. 41 Admiral Street, Port Jefferson Station, New York 11776. $50.

The advent of cladistic philosophy and methods has given systematics a more active, one might even say fundamental, role in the general framework of comparative biology. With emphasis on establishing only monophyletic taxonomic groups on the basis of synapomorphy and reflecting relationships in the form of cladograms, cladistics necessitates explicit formulation of hypotheses and results. Concurrent with the widespread acceptance of these tenets has been the development of a more critical protocol for character and character state elucidation, coding, and analysis. This shift toward greater empiricism, coupled with parsimony as the ultimate arbiter in cladogram selection, has certainly placed a burden on the systematist by requiring cladograms to represent character state distributions as accurately as possible.

Systematists are, however, constrained in their ability to construct cladograms by hand since the possible number of equally parsimonious cladograms rises dramatically with the addition of taxa and/or characters. Even a general attempt at manual construction will be precluded by the sheer number of characters traditionally recognized, for example, in arthropod or vertebrate groups.

The simultaneous evolution of computers and cladistics packages has seen a trend from relatively inaccessible mainframe programs to the large-scale distribution of microcomputer versions, such as **PHYLIP** and **PAUP**, developed by Joseph Felsenstein and David Swofford, respectively. Recently, a series of empirical comparisons have been reported for various mainframe (e.g., Luckow and Pimentel, 1985) and PC versions (e.g., Fink, 1986; Platnick, 1987, 1988, in press; see also Coddington, 1987). With such publicity, coupled with rumors of "this new version" or "that new program" about to be released, one gets the impression that we are in the midst of an event which will be of great benefit to systematists: a programming race to produce faster algorithms for finding all minimum-length trees, yet which are compatible on a variety of PC's. The latest contender in this race is **Hennig86,** version 1.5, developed by J. S. Farris (1988) for MS-DOS, IBM-compatible PC's. This review is not intended to make empirical comparisons of features or results from **Hennig86** with those, for example, from **PHYLIP** or **PAUP.** The most current information on these aspects has been prepared by Platnick (in press) as an update of his earlier analyses (Platnick, 1987).

Hennig86 is a surprisingly compact program (49K), yet extremely powerful and impressive in its tree-building capabilities, speed, and extensive tree/data-manipulating commands. Further, it does not require a math co-processor. The distribution disk contains three files: 1) **ss.com,** which is **Hennig86** per se; 2) **dox,** a command help file which can be accessed while working within the program; and 3) **peg,** a sample character state matrix. About 512K of RAM are required for its operation.

Hennig86 is fully interactive, providing a variety of cladogram and character editing features. Cladograms are generated by strict parsimony analysis. There is, however, no facility for implementing any form of Dollo or Camin-Sokal parsimony (ala **PAUP**; Swofford, 1985). Workers familiar with the **PHYSYS** mainframe program, developed by J. S. Farris and M. F. Mickevich (Mickevich and Farris, 1984), will find much of the operation of **Hennig86** very familiar. Those more familiar with any of the other available programs will probably come to fine the ease, logic, and (best of all) speed of **Hennig86** to be extremely gratifying.

The documentation for **Hennig86** might seem sparse, comprising only 15 text pages divided into 23 sections. This and its rather terse wording will likely be intimidating to some. Initial apprehensions aside, the user should find all instructions and examples quite comprehensible. Most importantly, the documentation is arranged such that descriptions of commands and internal program files in one section will usually have a direct bearing on how one or a series of commands in a later section can be successfully initiated. With each section dependent information-wise upon earlier sections, it is best to initially proceed through the documentation in sequence, sparing one the frustration of having to continually backtrack to determine why a command will not work. Also, working through each of the documentation's examples with the sample data matrix, **peg,** is quite helpful. Indeed, the documentation's idiosyncratic style is especially effective in prodding the user into experimenting with various commands and options, reducing the degree to which one might be inclined to simply view **Hennig86** as just another "black box" program. This makes **Hennig86** a must, not only for the established systematist, but for students as well. Several sections, however, should probably be expanded to include additional examples as well as further details about applications, interpreting results, and avoiding pitfalls.

Data matrix files can be entered into **Hennig86** as DOS or ASCII word-processing files. While the documentation describes the format for setting up a data matrix, users should examine the contents of **peg** for a good example. As noted earlier, **peg** is also a good sample data matrix with which to explore all available options. **Hennig86** can accept from 1 to 999 characters and 4 to 180 taxa. Character states must be integer coded. In the edition of 1.5 reviewed here (obtained in August 1988), the number of states for a character is limited to the range 0–9. A future update of 1.5 will extend this range to about 36 states (Farris, pers. comm.). Missing or unknown data are allowed, coded as "?" or "–." Characters can be differentially weighted, with weights ranging from 0 to 100 (the default value is 1; also see discussion below on successive approximations weighting). Multistate characters may be treated as additive ("ordered") or non-additive ("unordered"). In the additive form, transformation series can only be arranged linearly. Input of user-defined branching character state trees will be available in a later version for those who wish to explore this option. In the meantime, one can achieve the same end through a variety of additive coding methods (e.g., Pimentel and Riggins, 1987; O'Grady and Deets, 1987). Polymorphic characters per se cannot be designated. The **ccode** command enables the user to delete, weight, or change the additivity of characters for a given analysis. Unfortunately, there is no similar command for temporarily deleting taxa.

The documentation does not explicitly forewarn users of one minor detail which could cause initial problems. **Hennig86** always numbers input taxa and characters, internal program files, and all output listings in a consecutive manner starting with

0. Thus, for example, if one decides to input a data matrix with the fifth character weighted higher than the default value, this character must be referred to as character 4 in the weighting command.

Hennig86 offers a wide range of cladogram-calculating commands based on "exact" and "approximate" algorithms. When multiple minimum-length cladograms are generated, only unique topologies are retained, i.e., redundant cladograms with unsupported or zero-length branches are collapsed to show all polytomies. Cladograms must always be rooted by at least one outgroup taxon and there is the option of designating any number of additional taxa as secondary outgroups. A command for rerooting cladograms is also available (**reroot,** described below).

The "exact" algorithm, **ie** ("implicit enumeration"), will find all minimum-length cladogram(s), but its success may be dependent upon the number of final cladograms that are saved by a particular option and/or the amount of available memory. Based upon the particular option selected with **ie**, the number of final cladograms retained can be limited to 1, may go up to 100, or all available memory may be used. Because of the exhaustive search strategy performed by the **ie** command, its use may be prohibitive timewise due to size of the data set, amount of homoplasy, and/or size and speed of the computer's microprocessor. This is a matter of how long the user wants to tie up the machine, especially if it is not multi-tasking. The bottom line is that **Hennig86** does not discriminate on the basis of data size when it comes to how extensively one wants a search to be executed. But, even on my Toshiba laptop, with a 9.54 MHz 80C86-1 microprocessor, I have been able to run relatively large data sets very quickly.

There are two "approximate" algorithms, each with several options. The least effective of these, **hennig,** makes a single pass through the data, constructing one cladogram, which may not be of minimum length. Limited branch-swapping can be applied to this cladogram, but again, only a single cladogram is retained. The command, **mhennig,** constructs several initial cladograms, each by a single pass, but adds taxa in several different combinations, saving all minimum-length cladograms. Limited branch-swapping can be performed on these cladograms with **mhennig*.** For very large or messy data sets, the only feasible approach to obtaining optimal or near-optimal results in a timely fashion is **mhennig*** in combination with **bb** or **bb***. The **bb** command performs extended branch-swapping on all cladograms generated from **mhennig***, saving all cladograms it can find up to a limit of 100. The number of cladograms retained can be upgraded to the limit of available memory by using **bb*** . The efficiency of **mhennig*** with **bb*** to find all most parsimonious cladograms appears to be quite good (Platnick, 1988, pers. comm.).

If I were to order the search strategies from best to worst from the standpoint of finding as many minimum-length cladograms as possible, I would suggest the following: 1) **ie***, 2) **ie**, 3) **ie-** with **bb** or **bb***, 4) **mhennig*** with **bb** or **bb***. Similar suggestions are made by Farris (1988; see also Platnick, 1988 for comparisons of results). Again, the trade-off is the possibility of not finding all minimum-length cladograms for the benefit of a shorter run time.

Cladograms can be output in the form of branching diagrams or in parenthetical notation, and can be examined as output directly from the monitor and/or saved to a disk file. Upon obtaining results, one can select, with the **tchoose** command, a particular group of cladograms which can be further examined on the monitor or saved to a disk file using the **tsave** command. A particularly nice option when printing out cladograms is that branches can be displayed using extended ASCII symbols (the default), which could conceivably make diagrams of publishable quality. Moreover, cladograms are diagrammed such that taxa are placed on consecutive, single-spaced lines, considerably reducing the amount of printout.

Cladograms can be diagnosed to varying degrees with the **xsteps** command in conjunction with a series of specified options. Diagnoses can include, for example, length, consistency (CI) and retention indices (r_i) of each cladogram, CI and r_i of each character, number of steps required for each character on each cladogram, best and worst fits (CI and r_i) of each character for a set of cladograms, and all possible states at the nodes of each cladogram (i.e., hypothetical ancestral states).

The retention index, developed by Farris for **Hennig86**, is a measure of the ability of a character to function as a synapomorphy relative to the overall consistency of that character. The index is not described in the documentation and a formal description has not yet been published. The index is defined as $r_i = (h_i – s_i)/(h_i – l_i)$, where h_i is the largest number of steps possible for character i on any cladogram topology for a given set of taxa, l_i is the smallest number of possible steps, and s_i is the observed number of steps for the actual cladogram (Farris, pers. comm.). An r_i of 1 denotes a character which is completely consistent on the cladogram and with at least one state acting as a synapomorphy, whereas a value of 0 indicates unique character state changes limited only to terminal taxa. Values less than 1 and greater than 0 indicate some degree of homoplasy or reversal. Since the r_i is sensitive to the number of states acting as synapomorphies it will not always correspond to the CI.

In the event there are different equal-length transformation series for a particular character, including equal possibilities of reversal or parallelism, **Hennig86** will automatically list all possible states that that character can manifest at a given node. This is comparable to output from the **CSPOSS** command in **PAUP** (Swofford, 1985) but is presented in a more concise manner in **Hennig86**. A graphic representation of character state ambiguity is provided by the tree editor, **Dos Equis** (see below). All diagnostics are printed out as very compact tables which take up as little space as possible. This is not only a convenience when having to examine reams of printout, but, like the cladograms, makes for ease of examination directly from a monitor.

My only complaint is that there is no comparable listing for unique state changes (e.g., a reversal or loss) occurring in terminal taxa. Autapomorphies in terminal taxa can be detected by comparing the listing of total step changes for a character with that character's total number of steps at the nodes. A discrepancy indicates that terminal-taxon changes have occurred. One will then have to examine the original data matrix to find which taxa have uniquely derived the condition, then map the change on the cladogram(s). This is actually not much of an inconvenience, but should be watched when mapping character states onto cladograms. Another method of searching for terminal taxon changes is with the tree editor, **Dos Equis** (see below). Farris (pers. comm.) is in the process of making it easier to account for autapomorphies with the **xsteps** command.

The command, **nelsen,** calculates a Nelson consensus tree (Nelson, 1979; Nelson and Platnick, 1981) of all cladograms from a given tree file.

Hennig86 also allows for a successive approximations weighting procedure (Farris, 1969; see also Carpenter, 1988), which is effective in reducing the number of equal-length cladograms by an iterative series of a posteriori weightings. This has the effect of choosing the cladogram(s) with the most consistent (i.e., cladistically reliable) characters. Successive approximations weighting affords one the opportunity to reduce the number of cladograms that must be inspected. This might be useful if one is more concerned with getting patterns of relationship based on as few cladograms as possible without sacrificing character support, or deleting cladograms on the basis of a priori assumptions.

In this procedure, each character in the initial set of cladograms is assigned a weight, scaled between 0 and 10. Weights are calculated by the **xsteps** command with the **w** option as products of the highest CI and r_i values as determined from the best fits statistics. The data matrix is rerun with characters weighted accordingly. New weights are then calculated from the new cladogram(s), applied again to the original data, and rerun. The procedure is terminated at the point in which weights no longer change with each iteration, indicating cladogram topologies are not changing from one run to the next. Unlike **PHYSYS**, there is no command loop available to automatically switch weights and carry out each run to termination (Carpenter, 1988). The weighting command in **Hennig86**, however, makes the task so easy that this is hardly an inconvenience.

The weighting function in **Hennig86** differs from that in **PHYSYS** in that weights in the latter are calculated as the mean CI of each character (Carpenter, 1988). Since mean CI takes all values into consideration from all cladograms generated, it is probably a stronger weighting function than that in **Hennig86**. Carpenter (1988) reiterated the suggestion made by Farris (1969) that integer-coded, additive multistate characters should be recoded in additive binary form to avoid uneven weighting and effects of character state dependency. Additive binary coding effectively treats each state independently. This alternative is not feasible in the case of integer-coded, non-additive multistate characters since conversion to a non-additive binary form precludes determination of nodal conditions or transformation series (Farris, pers. comm.)

Hennig86 accepts the input of user-defined cladograms by use of the **tread** command. This has the utility, for example, of taking a published cladogram on which no character support has been shown (which is quite common) and optimizing characters from the original data matrix onto this topology. Subsequently, by using the tree editor, **Dos Equis** (described below), one may interactively examine, edit, and/or save results of any further manipulations. Another approach to the same problem, however, is to simply edit the cladogram(s) generated from the data matrix by **Hennig86**, using **Dos Equis**, to conform to the published topology. Discrepancies between published results and those found by **Hennig86** can be readily determined from the diagnostics output from **xsteps** or **Dos Equis**.

User-defined cladograms to be input into **Hennig86** (using the **tread** command) must be expressed in some form of parenthetical notation. Unresolved groupings are allowed and terminal taxa may be referred to by number or name. There is no need to balance all parentheses, especially in complex asymmetrical topologies. For example, the expression (((1 2)(3 4))(5 6)) can also be input as 1 2)(3 4))(5 6. **Hennig86** is very liberal in the types of symbols that are allowed to delimit groups, i.e., ( ), \ /, [ ], { }, and comma. When used with asymmetrical topologies, each symbol has a given priority level relative to all others such that one symbol will force symbols of equal or lower priority to balance out when read. The symbols shown above are ordered from lowest to highest priority. The amount of time and effort needed to input topologies is considerably decreased. Thus, the notation (0[(1 2)\3(4(5 6(7 8 is the same as 0((1 2)(3(4(5 6(7 8 and (0((1 2)(3(4(5 6(7 8)))))). One of the examples provided in the **Hennig86** documentation (Section 7: Tree Input) is in error; 0[1(2(3 4/6(7(8 9 is said to be equivalent to 0((1(2(3 4)))(6(7(8 9)))). The first expression will not designate a sister-group relationship between (1 2 3 4) and (6 7 8 9). The abbreviated notation should actually be something like 0[1(2(3 4][6(7(8 9.

Hypothetical (nodal) ancestors can be specified in notation as a number preceded by a period (e.g., .0 or .2), with descent from an ancestor denoted by the connection sign "−." Large or complex clades can then be split apart by use of the comma as a delimiter (see above). This has the utility of making potentially unwieldy expressions easier to handle. For example, the expression .0\.1, .2, .0–0\1 2, .1–3\4 5, .2–6\7, states initially that ancestors .1 and .2 are descended from .0, establishing the sister group (3 4 5)(6 7). The entire grouping is therefore the same as 0\1 2,[3\4 5]\6 7 or (0(1 2)(3(4 5)(6 7))).

The process of compressing (or collapsing) zero-length or unsupported branches to polytomies during cladogram construction in **Hennig86** can also be applied to a set of user-defined cladograms using the **xsteps** command with the **u** option. Say, for example, the topologies (0(1(2(3 4)))) and (0(1(3(2 4)))) are input. If there is no character support for either (3 4) or (2 4), these are compressed to a polytomy resulting in a single unique cladogram: (0(1(2 3 4))). Since **Hennig86** does not generate redundant cladograms with unsupported branches, when would one need to worry about compressing cladograms? Both **PHYLIP** and **PAUP** (prior to version 3.0) will generate only fully dichotomous cladograms, including all possible (and redundant) fully resolved topologies for polytomous conditions. For the purposes of comparing only unique cladograms produced by either program, all cladogram topologies can be input into **Hennig86**, compressed, and examined. Often this will substantially reduce the number of cladograms that must be examined and affords easy comparison with cladogram(s) generated from the same data set by **Hennig86**.

A cladogram or set of cladograms can be rerooted using a different outgroup by the **reroot** command. If a new outgroup has been designated, invoking the **reroot** command will produce all new and unique cladograms based on this new outgroup. The new cladogram(s) can then be diagnosed with the **xsteps** command. Similarly, cladograms can be rerooted, examined, and diagnosed from the tree editor, **Dos Equis**, described next.

Hennig86 provides a very nice interactive tree editor, **Dos Equis**, which has capabilities very similar to those seen in the program **MacClade**, developed by Wayne and David Maddison for Macintosh computers. Cladograms generated by **Hennig86** or other user-defined cladograms can be used. **Dos Equis** is entered, not surprisingly, with the command, **xx**. A single cladogram from the program's current internal tree file is displayed until another cladogram is chosen. The states for a particular character are indicated for each terminal taxon and at all nodes. As mentioned earlier, in the event that several equal-length transformation series are possible for a given character, all possible nodal conditions are presented. Additional diagnostic data are shown below the cladogram and include total cladogram length, which character is being displayed, current weight of that character, if the character is additive or non-additive, if it has been used in cladogram construction (active) or not (non-active), and the number of steps required by that character to fit on the cladogram.

Editing with **Dos Equis** is very straight-forward, with onscreen editing controls displayed in a concise manner. As is typical of **Hennig86**, a minimal number of key strokes are required to initiate commands. Cladogram topology modifications include moving terminal branches or clades, and deleting nonterminal branches (clades). Upon making such changes the user is updated as to total cladogram and character length. Unfortunately, a single terminal taxon cannot be deleted. Possible character modifications include changing weights, activity, and additivity. Again, the user is updated on the changes incurred with these editing procedures with regard to total character length. All desired changes can then be saved, or else the user can exit directly from **Dos Equis**.

While one can move or rotate branches or taxa, and see the effects directly, the same immediate results are not achieved with character modifications. For example, if a character is made inactive (i.e., essentially deleted for purposes of cladogram construction), this is indicated on the update, but the character is still shown on the cladogram as though it were active. In order to see what effect this change actually has on the cladogram topology, one must save the modified cladogram and character settings and rerun this new data set.

In all, **Hennig86** meets virtually all the criteria one would expect in a cladistics program. Its small size, low cost, and compatibility make it readily accessible to a wide audience. The few problems pointed out here are certainly minuscule compared to the overall benefits provided. Obviously, one's acceptance and use of a particular program is an indication that it meets, at least minimally, the user's expectations, which might include 1) ease of interaction, 2) ability to handle a variety of data sets of different sizes, 3) relatively good speed in analyzing data, 4) a variety of search strategies, 5) receiving concise and accurate results, and 6) being able to easily handle and interpret output.

No doubt with the introduction of new programs, and revisions of old ones, users will begin to weigh differences and similarities based on their own expectations. Differences of opinion will probably develop mainly as a function of these expectations, as well as due to theoretical and research proclivities, and associated ad hoc assumptions deemed allowable. Biases aside, systematists should definitely take the time to assess for themselves what they perceive to be the strengths and weaknesses of **Hennig86** as it pertains to their own research and teaching.—*Kirk Fitzhugh, Department of Invertebrates, American Museum of Natural History, New York, New York 10024.*

LITERATURE CITED

Carpenter, J. M. 1988. Choosing among multiple equally parsimonious cladograms. Cladistics 4:291–296.

Coddington, J. A. 1987. The sixth annual meeting of the Willi Hennig Society. Cladistics 3: 178–184.

Farris, J. S. 1969. A successive approximations approach to character weighting. Syst. Zool. 18:374–385.

Farris, J. S. 1988. **Hennig86** reference. Documentation for version 1.5.

Fink, W. L. 1986. Microcomputers and phylogenetic analysis. Science 234:1135–1139.

Luckow, M. and R. M. Pimentel. 1985. An empirical comparison of numerical Wagner computer programs. Cladistics 1:47–66.

Mickevich, M. F. and J. S. Farris. 1984. **PHYSYS** documentation.

Nelson, G. 1979. Cladistic analysis and synthesis: principles and definitions, with a historical note on Adanson's "Familles des plantes" (1763–1764). Syst. Zool. 28:1–21.

Nelson, G. and N. I. Platnick. 1981. Systematics and Biogeography; Cladistics and Vicariance. Columbia University Press, New York.

O'Grady, R. T. and G. B. Deets. 1987. Coding multistate characters, with special reference to the use of parasites as characters of their hosts. Syst. Zool. 36:268–279.

Pimentel, R. M. and R. Riggins. 1987. The nature of cladistic data. Cladistics 3:201–209.

Platnick, N. I. 1987. An empirical comparison of microcomputer parsimony programs. Cladistics 3:121–144.

Platnick, N. I. 1988. Programs for quicker relationships. Nature 335:310.

Platnick, N. I. In press. An empirical comparison of microcomputer parsimony programs, II. Cladistics.

Swofford, D. L. 1985. **PAUP** (phylogenetic analysis using parsimony). Documentation for version 2.4. Illinois Natural History Survey, Champaign.

J. New York Entomol. Soc. 97(2):241–242, 1989

TWO NEW TRUE BUG CATALOGS

Catalog and Bibliography of the Leptopodomorpha (Heteroptera).—R. T. Schuh, B. Galil, and J. T. Polhemus. 1987. Bulletin of the American Museum of Natural History 185:243–406. $10.65.

For most biologists, and especially for museum curators, taxonomists, and biogeographers, the most important source of reference is a worldwide catalogue. Unfortunately, few people want to undertake the tedious and time-consuming work involved in making such a catalogue. The present volume is therefore received with great enthusiasm.

For many higher groups of Heteroptera or true bugs, the only worldwide catalogue is still that of L. Lethierry and G. Severin (1893–1896). Needless to say, this catalogue is hopelessly outdated. The "General Catalogue of the Hemiptera," initiated in 1927 (Editors G. Horvath and H. M. Parshley) was never completed as far as the Heteroptera is concerned. In fact, only two heteropteran families were ever treated, the Mesoveliidae by G. Horvath and the Pyrrhocoridae by R. F. Hussey (both in 1929).

Fortunately, other worldwide catalogues have appeared, foremost among them the impressive catalogues on the Miridae by J. C. M. Carvalho (1957–1960), on the Lygaeidae by J. A. Slater (1964), and on the Tingidae by C. J. Drake and F. A. Ruhoff (1965). Nevertheless, most families of the Heteroptera, including such large and important groups as the Reduviidae, Coreidae, and Pentatomidae, have not been adequately catalogued.

The present "Catalog and Bibliography of the Leptopodomorpha" covers one of the smallest of the heteropteran infraorders with less than 300 described species. The infraorder comprises the shore bugs and allied groups. Most species inhabit damp soil close to water, either fresh or saline. A few species are intertidal marine. The